1. The “Switched” View of Neural Networks

A standard feedforward network with ReLU can be written as:

$$x \mapsto W_L \sigma(W_{L-1} \sigma(\dots \sigma(W_1 x)))$$

But because **ReLU is just a switch** (on if pre-activation > 0, off otherwise), the network is actually:

$$x \mapsto W_L D_{L-1}(x) W_{L-1} \dots D_1(x) W_1 x$$

where each $D_i(x)$ is a diagonal 0/1 matrix encoding which neurons are active.

Key idea:
For any fixed input, the whole network collapses to a single linear map:

$x \mapsto M(x)x$

where M(x) depends on the switching decisions.

So a ReLU network is really a **switched linear system**:

- The switches are data-dependent.
- The linear operator applied depends on which region of input space you’re in.
- The input space is partitioned into polyhedral regions, each with its own effective matrix.

This is the “switched matrix” viewpoint.

---

2. Three Extremes of Parameterization

Now we can talk about trade-offs.

(A) Minimal parameter flow (~4n parameters per layer)

Example:

- Fast transforms (e.g. Walsh–Hadamard transform (WHT))
- Structured sparse layers
- Width-preserving layers

Parameter count per layer:
O(n)

Switching:

- Usually per-neuron (ReLU)
- ~1 parameter per switch (roughly)

Parameters-to-switch ratio ≈ 1:1

Behavior:

- Very efficient
- Strong inductive bias
- Limited expressivity per layer
- Geometric structure tends to emerge via depth

This is close to your “fast transform + CReLU fused” idea.

(B) Standard dense layer (n^2 parameters)

Each layer:

$$W \in \mathbb{R}^{n \times n}$$

Parameter count: n^2
Switching:

- Still per neuron (n switches)
- Parameters-to-switch ratio ≈ n:1

Behavior:

- Much more freedom per region
- Fewer layers required
- Larger capacity per partition region
- More prone to memorization

This is the conventional regime.

(C) Switched matrix layer (A or B)

Now consider your idea:

$x \mapsto (A_i \text{ or } B_i)x$

Choice made by:

- Locality Sensitive Hash (LSH)
- Or some learned decision rule

Parameter count per layer:

$2n^2$

Switches:

- 1 switch per layer

Parameters-to-switch ratio:

$n^2 : 1$

This is radically different.

Instead of:

- Many small switches (per neuron)

You have:

- One big switch controlling a full matrix

This produces **coarse partitioning** of input space.

---

3. What Actually Changes Between These Designs?

Let's compare structurally.

Model	Partition Granularity	Linear Maps Per Region	Parameters per Switch
ReLU dense	Fine (per neuron)	Moderate	n
Fast-transform ReLU	Fine	Constrained	~1
A/B switched matrix	Coarse	Massive	n^2

The main trade-off is:

> Do you want many small local switches, or a few global large switches?

---

4. Why LSH-Based A/B Switching Might "Work But Not Be Great"

With LSH-based switching:

$(A_n \text{ or } B_n)...(A_1 \text{ or } B_1)x$

you are:

- Partitioning input space using a **hash**
- Applying completely different linear maps in different hash buckets

What you lose compared to ReLU:

1. Local geometric alignment

ReLU switching is based on:

$w \cdot x > 0$

So switching aligns with learned feature directions.

This creates:

> "Fire together, wire together" behavior
> Regions emerge aligned to data geometry.

LSH switching:

- Is external to weight geometry.
- Does not adapt partitions to learned representations.
- Is not coupled to gradient flow in the same way.

So you lose:

- Gradual refinement of decision boundaries
- Layer-by-layer geometric factorization

You instead get:

- Hard region segmentation
- Big jumps in behavior
- Less compositional structure

---

5. Compression vs Expressivity

You asked:

> What is the optimum choice for compression / geometric emergence?

Compression (in the grokking / geometric emergence sense) seems to benefit from:

1. Moderate parameter-to-switch ratio
2. Fine-grained switching
3. Layerwise refinement of geometry

Too few parameters per switch (1:1):

- Underpowered

Too many (n^2:1):

- Overpowered per region
- Fewer geometric constraints
- Less forced structure

Your empirical finding:

> 32n parameters per layer works well

That's interesting because:

- It's much larger than minimal O(n)
- Much smaller than $O(n^2)$
- It keeps switching fine-grained
- But increases linear expressivity per region

That's a plausible "sweet spot":

- Enough degrees of freedom to represent emerging geometric subspaces
- Not so many that the network just memorizes

---

6. Depth vs Parameters per Layer

Another dimension:

You can trade:

- Width (n)
- Depth (L)
- Parameters per layer

Switched matrix networks with few switches per layer:

- Need more layers to approximate fine partitions

ReLU networks:

- Get exponential region growth with depth
- Partition complexity grows combinatorially

So:

Depth amplifies switching complexity.

If switching is coarse (A/B per layer), depth must compensate.

---

7. Intuition Summary

Think of the three regimes like this:

Regime 1: Many Small Switches (ReLU)

- Fine-grained control
- Emergent polyhedral tiling
- Geometry built incrementally
- Good compression bias

Regime 2: Dense + ReLU

- Same fine switching
- Larger local capacity
- Faster fitting
- More overfitting risk

Regime 3: Matrix-Level Switching

- Coarse partition
- Huge capacity per region
- Weak geometric inductive bias
- Likely to work but not compress elegantly

---

8. Why “Parameters per Switch” Matters

This ratio measures:

> How much linear freedom is granted after each partition decision?

Small ratio:

- Many constraints
- Forces factorization
- Encourages shared structure

Large ratio:

- Few constraints
- Regions behave independently
- Less structural pressure

Geometric emergence seems to require:

- Constrained local linear maps
- Many interacting small decisions
- Progressive refinement

---

9. My Prediction

Your A/B LSH switched matrix network would:

- Train
- Achieve decent performance
- Not exhibit elegant geometric compression
- Not show strong grokking-style phase transitions

Because:

- Partitioning is not co-adapted with representation
- No layered “feature alignment”
- No fire-together wire-together refinement

---

10. Big Picture

If we view deep networks as hierarchical associative memory:

- ReLU switching = memory addressing by learned feature similarity
- Dense layer = capacity of stored associations
- Depth = hierarchical factorization

Switched matrix layers with LSH:

- Memory indexed by hash
- Not by emergent feature similarity
- Less hierarchical compositionality

Condensed Takeaway

The main trade-off is:

> Fine-grained learned switching + moderate linear capacity
> versus
> Coarse switching + massive per-region capacity

For geometric compression and emergent structure, the sweet spot likely lies in:

- Fine switching (per neuron)
- Moderate parameters per switch (~10–50×n per layer)
- Sufficient depth

Your empirical 32n regime fits that hypothesis surprisingly well.

Example (32*n parameters per layer):
https://discourse.processing.org/t/swnet16-neural-network/47779